Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2012) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

New Challenges in Systems Engineering and Architecting
Conference on Systems Engineering Research (CSER)
2012 – St. Louis, MO
Cihan H. Dagli, Editor in Chief
Organized by Missouri University of Science and Technology

Effectiveness of kanban approaches in systems engineering within rapid response environments

Richard Turner[a*], Dan Ingold[b], Jo Ann Lane[b], Raymond Madachy[c], David Anderson[d]

[a]*Stevens Institute, Hoboken, NJ, 07030, USA*
[b]*University of Southern California, Los Angeles, CA, 90089, USA*
[c]*Naval Postgraduate School, Monterrey, CA, 93943, USA*
[d]*David J. Anderson Associates, Seattle, WA, 98113, USA*

Abstract

In the last few decades, system contexts have multiplied, and the speed of change in both needs and solutions has accelerated. This has led to an inherent loss of determinism—requirements are less tangible, more evolving, and sometimes emergent, and systems are both complex and constantly adapting. Engineering principles involving agility and leanness have been adopted to address non-determinism in software systems. This paper examines one of those approaches, kanban (pull) scheduling techniques, to determine its applicability to systems and software engineering in a rapid response environment. The paper describes work in progress defining a general systems engineering kanban approach, a specific kanban process for rapid response, and the validation of that process via simulation.

Keywords: Agile systems engineering; lean systems engineering; kanban; rapid response; scheduling; systems engineering management

1. Introduction

Traditional systems engineering developed half a century ago, primarily driven by the challenges faced in the aerospace and defense industries. In the last few decades, system contexts have multiplied, complexity has grown exponentially, and the speed of change in both needs and solutions has accelerated. In rapid or continuous deployment environments, where requirements are not precise and can change or emerge quickly, traditional systems engineering has often failed to perform its tasks within the available schedule

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Effectiveness of kanban approaches in systems engineering within rapid response environments	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stevens Institute of Technology,Castle Point,Hoboken,NH,07030**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In the last few decades, system contexts have multiplied, and the speed of change in both needs and solutions has accelerated. This has led to an inherent loss of determinism?requirements are less tangible, more evolving, and sometimes emergent, and systems are both complex and constantly adapting. Engineering principles involving agility and leanness have been adopted to address non-determinism in software systems. This paper examines one of those approaches, kanban (pull) scheduling techniques, to determine its applicability to systems and software engineering in a rapid response environment. The paper describes work in progress defining a general systems engineering kanban approach, a specific kanban process for rapid response, and the validation of that process via simulation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

and resource bounds [1, 2]. Clearly, new and flexible methods, processes and tools are required for effective systems engineering in these environments.

Engineering principles involving agility and leanness have been adopted to address non-determinism in software systems. However, integrating these agility and leanness concepts into the systems engineering workflow has proven difficult. Leveraging work done in earlier research [3, 4], agile and lean practice research [5, 6, 7], and including new experience with lean approaches [8, 9], we are investigating the use of flow-based pull scheduling techniques (kanban systems) in a rapid response development environment.

2. Kanban systems in knowledge work

A kanban (signal card) approach provides a visual means of managing the flow within a process. The signal cards are created to the agreed capacity of the process and one card is associated with each piece of work. Here, work can mean the creation of a part, the integration of a part into an assembly, the completion of a particular analysis process, or whatever bounded and completeable task you wish to track through the process. Once all of the cards have been associated, no more work in that process can begin until some piece of work is completed and the card becomes available. An often used example of a simple kanban is the use of a limited number of tickets for entry into the Japanese Imperial Gardens [8]. The fundamental idea is to use visual signals to synchronize the flow of work with process capacity, limit the waste of work interruption, minimize excess inventory or delay due to shortage, prevent unnecessary rework, and provide a means of tracking work progress.

In knowledge work, the components of production are ideas and information [10, 11]. In software and systems, kanban systems have evolved into a means of smoothing flow by balancing work with resource capability. The concept was extended to include the limiting of work in progress according to capacity. Work cannot be started until there is an available appropriate resource. In that way, it is characterized as a "pull" system, since the work is pulled into the process rather than "pushed" via a schedule.

A kanban system is a visually monitored set of process steps, each step with its own queue and set of resources, that add value to work units that flow through them. The fact that queues are included in the system allows costs of delay and other usually invisible aspects of scheduling to be front and center in decision making. Queues also provide a vast body of experience and underlying science from the queuing theory discipline. Control of the process is generally maintained through *batch size*, *Work in Progress (WIP)* limits and *Classes-of-Service* (COS) definitions that prioritize work with respect to risk.

The visual representation of the work is critical to kanban success, because it provides immediate understanding of the state of flow through the set of process activities. This transparency makes apparent process delays or resource issues and enables the team to recognize and react immediately to resolve the cause. Kanban is also an embodiment of the continuous improvement concept (kaizen) Flow is measured and tracked through statistical methods that provide insight into needed changes in the control parameters to tune and improve the system. Measures also provide a good handle for effectiveness comparison.

WIP is partially-completed work, equivalent to the manufacturing concept of parts inventory waiting to be processed by a production step. WIP accumulates ahead of bottlenecks unless upstream production is curtailed or the bottleneck resolved [12]. WIP in knowledge work can be roughly associated to the number of tasks that have been started and not completed. *Limiting WIP* is a concept to control flow and enhance value by specifically limiting the amount of work to be assigned to a set of resources (a WIP Limit). WIP limits accomplish several goals: they can lower the context-switching overhead that impacts individuals or teams attempting to handle several simultaneous tasks; they can accelerate value by completing higher value work before starting lower value work; and, they can provide for reasonable resource work loads over time.

Using *small batch sizes* is a supporting concept to WIP to further limit rework and provide flexibility in scheduling and response to unforeseen change. Smaller batch sizes even out the process flow and allow downstream processes to consume the batches smoothly, rather than in a start-and-stop fashion that makes

inefficient use of resources. The move from "one step to glory" system initiatives to iterative, deployable increments is an example of reducing batch size. Incremental builds and ongoing, continuous integration also approximate the effect of small batch sizes.

3. Rationale for applying kanban to systems engineering

Systems engineering has struggled integrating in rapid-response environments, partly because it tends to operate in a broader, more holistic scope. In rapid response, the scope is often narrowed by the time scale and holistic thinking is perceived as less valuable. The idea of using a pull system for systems engineering is an attempt to draw that holistic thinking into the rapid development rather than lay it on top of the activities. This requires an entirely different way of viewing systems engineering. The concept we are researching would allow systems engineering to provide the holistic thinking as part of the services it provides each individual project. Certain SE activities are ongoing, system–level activities (e.g. architecture, environmental risk management). Others are specific to individual projects (e.g. trade studies, interface management). However, SE can be opportunistic in providing its cross-project view and understanding of the larger environment when servicing a project. It could also add or modify tasks within the project to ensure that an identified issue or external change is handled.

3.1. More effective integration and use of scarce systems engineering resources

Using kanban and applying a model of SE based on continuous activities and taskable services is a value-based way to prioritize the use of scarce SE resources across multiple projects. The value function within the next-work selection process can be tailored to provide efficient and effective scheduling that maximizes the value provided by the resource based on multiple, system-wide parameters.

3.2. Flexibility

SE activities are generally designed for pre-specifiable, deterministic (complete and traceable) requirements and schedules. There is often an overdependence on unnecessary formal ceremony and fairly rigid schedules. Using cadence rather than schedule can provide efficient SE flow with minimal planning. We believe that the CoS concept not only handles expedite and date-certain conditions, but also supports cross-kanban synchronization. Even though the planning is constant and the selection of the next piece of work to do asynchronous, we believe CoS and cadence provide a sufficient level of predictability where necessary.

3.3. Visibility and coordination across multiple projects

In highly concurrent engineering tasks, a kanban approach provides a means of synchronizing activities across mutually dependent teams by limiting their progress according to the degree of data completeness and maturity (risk of change). Kanban also provides an excellent way to show where tasks are and the status of work-in-progress and queued or blocked work.

3.4. Low governance overhead

Kanban systems don't require major changes in the way work is accomplished or organizational structures (such as Scrum). Kanban systems can be set up in individual projects and allowed to evolve into more effective governance. Even the systems engineering resource kanban can be implemented with very little organizational impact. Practitioners make most decisions using parameters set by management (e.g. WIP limits). Issues are usually identifiable from walking the board and so clear to all who take part

in the kanban, including management. Metrics are inherent to the system, clearly identify problems, and track improvements. Most problems tend to be self-correcting.

4. Research underway

4.1. Early model of a kanban system

In Figure 1 and Table 1, we present an early model of a kanban system. We intend that this model be recursive at many levels to allow for complex implementations. While we currently believe tasks and their associated parameters coupled with the visual representation of flow are sufficient, we may introduce new concepts to enable better communications and synchronization between kanban systems.

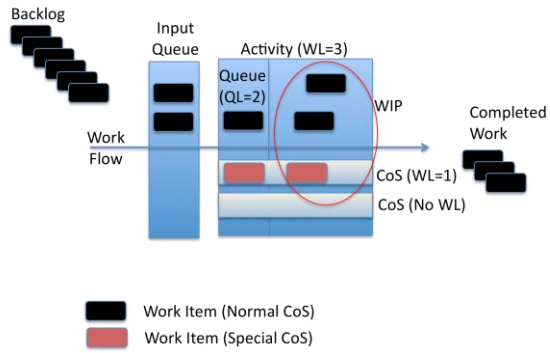

Figure 1. Kanban Model

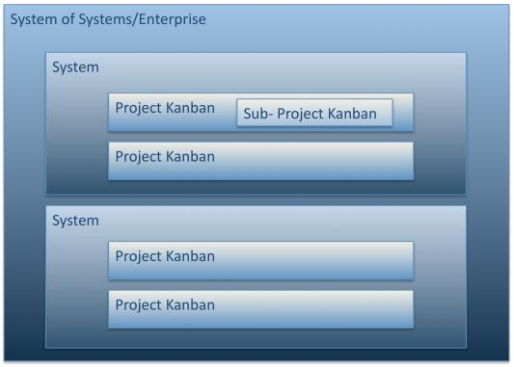

Figure 2. Kanban hierarchy

4.2. Kanban as part of rapid response workflow

Kanban fits into the rapid response workflow within both the SE and development specific activities. Assume that a rapid response begins with a customer need (or set of needs). The first level assessment of the backlog identifies relationships (if any) between various needs and candidate options for meeting those needs. The team looks at the full spectrum of options from process to material. For those options that have a technical system solution (or part of the solution is changes to existing systems/new system), further analysis is needed. This is a point where the lead SE might develop a statement of "needed" work/tasks (e.g. trade studies, prototypes, surveys, risk assessments). Developers and SEs would all be resources for this work, and one or more kanban systems could be established according to identified risks or other attributes. Other work could be ongoing and not a part of the kanban systems. These are things that can be done in parallel, but eventually need to come together to support a specific set of decisions to determine the next level of development tasks. New kanban systems may be created with work flowing to and from existing projects or tasks. Such a hierarchy is characterized in Figure 2. Because kanban systems are easy to implement and work relatively independently, they can be created, modified and removed on the fly.

4.3. Simulation of concept implementations

In order to evaluate the effectiveness of possible kanban systems, we are building a simulation infrastructure with a number of different capabilities. Several approaches to the simulation and evaluation are described in the next sections.

Table 1. Kanban System Definitions

Unit of Work	Determines the approximate size of work
Work Item	The item controlled in the kanban system; essentially, the kanban carrier
Work Flow	A collection of work items that enter, are performed, and exit the kanban system
Backlog	A set of work items from which the next work item is selected when a kanban system Input Queue contains less work than its limit (available space). Backlog is maintained by the kanban system, but the selection of next work is generally performed by the customer or other upstream entity.
Cycle	A time increment used to identify the amount of work done within an activity. Often, but not necessarily, a workday.
Cadence	The rhythm of the production system, usually some specific number of cycles. Not necessarily an iteration. Kanban still allows for iterations but decouples prioritization, delivery and cycle time to vary naturally according to the domain and its intrinsic costs.
Activity	Value-adding work that can be determined as complete. Includes: activity queue, a set of resources, and a WIP Limit
Resource	An agent for accomplishing work; may be generic or have specialized expertise. Includes: expertise-effectiveness pair(s), where effectivenes is in Units of Work performed per Cycle. Usually associated with a specific activity, but may be shared across activities.
Algorithm for Selecting Next Work Item	Rules for selecting the next work item from a queue when an activity has less work than its WIP limit; Can range from simple FIFO to complex, multiple kanban-system, multi-factor method considering shared scarce resources and multiple cost/risk factors.
Input Queue	A queue that holds the selected backlog items for the kanban system to process. This queue hs a strict size limit based on the overall capacity of the kanban system. As resources become available, backlog items are selected and added to the Input Queue. Selection of backlog items is controlled by the customer or entity just upstream from the kanban system, but control may be delegated to the kanban system.
Activity Queue	Holds work items for the Activity until WIP is under the limit. Includes: An algorithm for selecting next work item, optional size limit (number of work items)
Value Function	Estimates the current value of a work item for use in the selection algorithm. Can be simple or complex.
Class of Service	Provides a variety of handling options for work items. May have a corresponding WIP limit for each activity to provide guaranteed access for work of that class of service. CoS WIP limit must be less than the activity's overall WIP limit. Examples are expedite, date-certain and normal.
WIP Limit	Limit of work items allowed at one time within an activity
Visible Representation	A common, visual indication of work flow through the activities; Often a columnar display of activities and queues. May be manual or automated. Shows status of all work-in-progress, blocked work, WIP limits
Metrics	Includes cumulative flow charting and average transit (lead) time

4.3.1. System dynamics continuous modeling

Systems dynamics is a continuous-domain simulation method that has been widely used to model software development processes of all types. A systems dynamics model is based on the principals of flow—that the levels of various quantities to be measured accumulate over time, as determined by the rates of inflow and outflow, which themselves may be affected by feedback. In software development processes, the quantities to be measured may include developed software artifacts, staffing levels, defect rates, among many other parameters.

4.3.2. Discrete event simulation

Discrete-event simulations measure the inter-arrival rate of discrete work items at queues and the processing time in each production step. It can be a powerful tool to determine the effects of work item sizing, COS decisions, and WIP limits on the variability of processing time within and transit delay through the interdependent kanban systems.

4.3.3. Agent-based simulation

Rather than modeling the arrival rates of information at service queues, agent-based modeling (ABM) models the individual actors within a system and their behaviors, from which may arise queue-like behavior [13]. We believe ABM is a powerful approach for modeling kanban systems, by extrapolating system performance from individual behaviors, rather than from an assumed or idealized process model.

4.3.4. Hybrid simulation approaches

The most descriptive view of kanban performance will likely come through use of a combination of these simulation approaches. Continuous flow models provide a quick, macro-level view of system behavior. Discrete-event simulations quantify system throughput and determine the precise effects of process model choices. Agent-based modeling shows the systemic effects of individual performance. Together these approaches can be used to guide the design of an effective kanban-based approach.

5. Discussion on work to date

As of this writing, we have delved deeper into the relationships within kanban systems and established a hierarchy of scope that we believe matches the rapid response environment we intended to study. The model of the kanban system has been joined with the concept of SE as a taskable service, and the service definitions are under development. Several systems dynamics models have been built, but none seemed sufficient to act as the core of the simulations facility. We have built one agent-based model, and while complex, seems much more likely to provide the required simulation flexibility.

References

1. NDIA-National Defense Industrial Association (2010). Top Systems Engineering Issues In US Defense Industry. Systems Engineering Division Task Group Report, http://www.ndia.org/Divisions/Divisions/SystemsEngineering/Documents/Studies/Top%20SE%20Issues%202010%20Report%20v11%20FINAL.pdf. September, 2010.
2. Turner, Richard, Shull F., et al (2009a) "Evaluation of Systems Engineering Methods, Processes and Tools on Department of Defense and Intelligence Community Programs: Phase 1 Final Technical Report," Systems Engineering Research Center, SERC-2009-TR002, September 2009.
3. Turner, Richard, Shull F., et al (2009b) "Evaluation of Systems Engineering Methods, Processes and Tools on Department of Defense and Intelligence Community Programs: Phase 2 Final Technical Report," Systems Engineering Research Center, SERC-2010-TR004, December 2009.
4. Turner, Richard and Wade, J. (2011). Lean Systems Engineering within System Design Activities, Proceedings of the 3rd Lean System and Software Conference, May 2-6, 2011, Los Angeles, CA.
5. Boehm, Barry and Turner, Richard (2004). Balancing Agility and Discipline: A Guide for the Perplexed. Boston, MA: Addison Wesley.
6. Larman C. and Vodde, B. (2009). Scaling Lean & Agile Development. Boston, MA: Addison Wesley.
7. Poppendiek, Mary. (2007). Implementing Lean Software Development: Boston, MA: Addison Wesley.
8. Anderson, David. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Sequim, WA: Blue Hole Press
9. Reinertsen, Donald G. (2010). The Principles of Product Development Flow. Redondo Beach, CA: Celeritas Publishing.
10. Poppendieck, Mary, and Tom Poppendieck. (2003). Lean Software Development: An Agile Toolkit. The Agile Software Development Series. Boston: Addison-Wesley.
11. Morgan, James M, and Jeffrey K Liker. (2006). The Toyota Product Development System: Integrating People, Process, and Technology. New York: Productivity Press.
12. Goldratt, Eliyahu M., and Jeff Cox. (2004.) The Goal: a Process of Ongoing Improvement. Great Barrington, MA: North River, 2004.
13. Heath, B. et al. (2009.) A survey of agent-based modeling practices (january 1998 to july 2008). Journal of Artificial Societies and Social Simulation. 12:4 2009.